

Machine Learning Evaluation & Supervised Learning

Regression



Rezki Trianto

Data guy who spent the last 7 years to work around data analytics and machine learning. Working in a unicorn company with various domain expertise to work with, from an e-commerce, OTA, and ride-hailing company.



Rezki Trianto

[linkedin.com/in/rezkitrianto](https://www.linkedin.com/in/rezkitrianto)

Education Background



2010-2014
Bachelor Degree
Computer Science



2015-2017
Master Degree
Computer Science

Rezki Trianto

5+ years experience in Data Science & Analytics

Hands On Required

Hands - On : 2. Regression

Klik disini untuk
mengakses folder Hands
On

Topik Supervised Learning: Regression

- ☐ Review Supervised Learning
- ☐ Linear Regression
- ☐ Multivariate Regression
- ☐ Feature Importance
- ☐ Asumsi Linear Regression
- ☐ Hands On Linear Regression - Part 1
- ☐ Hands On Linear Regression - Part 2
- ☐ Regularization
- ☐ Hands On - Regularization
- ☐ Hyperparameter Tuning
- ☐ Hands On - Hyperparameter Tuning
- ☐ Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



☐ Review Supervised Learning

☐ Linear Regression

☐ Multivariate Regression

☐ Feature Importance

☐ Asumsi Linear Regression

☐ Hands On Linear Regression - Part 1

☐ Hands On Linear Regression - Part 2

☐ Regularization

☐ Hands On - Regularization

☐ Hyperparameter Tuning

☐ Hands On - Hyperparameter Tuning

☐ Hands On - Other Regression Algorithm

Masih ingat jenis-jenis task ML?

Supervised Learning

- Tersedia data + target
- Klasifikasi, **regresi**

Unsupervised Learning

- Tersedia data tanpa target
- Clustering, representation learning

Reinforcement Learning

- Data tidak tersedia (???), hanya aturan
- Melatih 'agen' dalam suatu 'task'

** diluar scope Bootcamp*

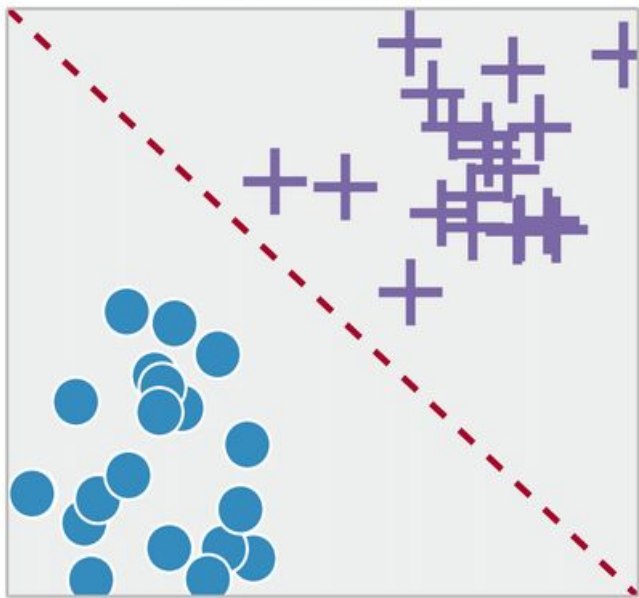
Supervised Learning

Revisit the idea

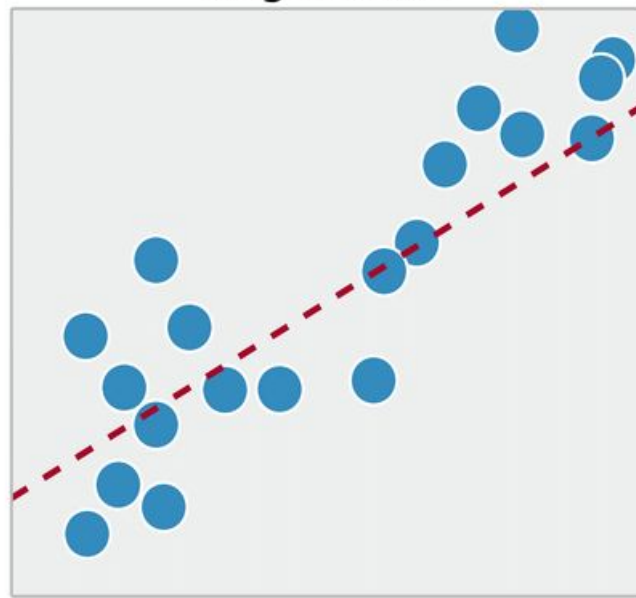
Melakukan prediksi dengan sebuah aturan.
Mempunyai data input (feature) dan label target

Jenis Supervised Learning

Classification



Regression



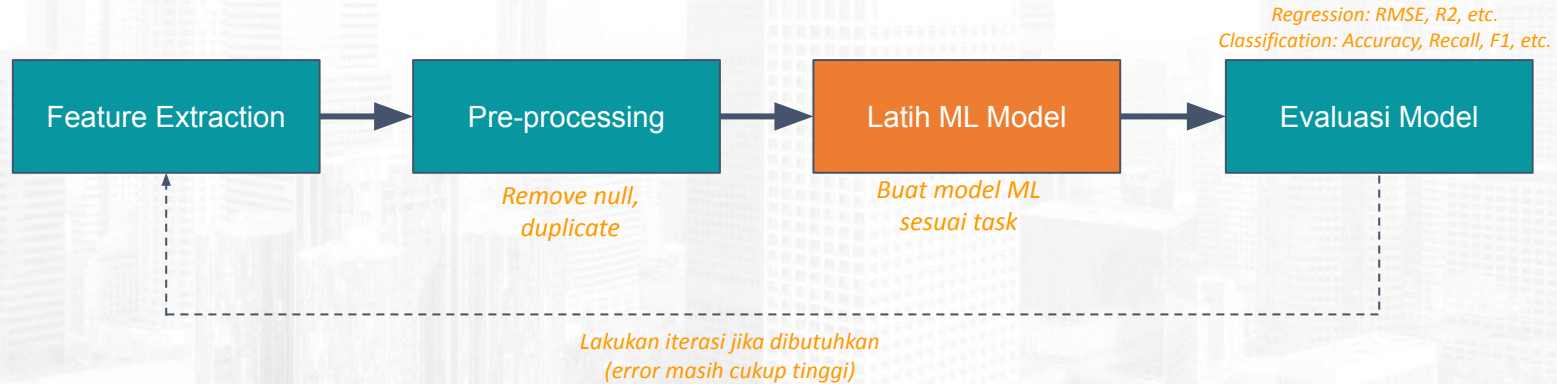
Target Label bertipe numerik

Target

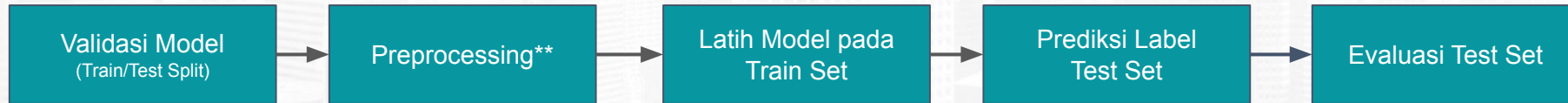
Feature

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above
221900	3	1	1180	5650	1	0	0	3	7	1180
538000	3	2.25	2570	7242	2	0	0	3	7	2170
180000	2	1	770	10000	1	0	0	3	6	770
604000	4	3	1960	5000	1	0	0	5	7	1050
510000	3	2	1680	8080	1	0	0	3	8	1680
257500	3	2.25	1715	6819	2	0	0	3	7	1715
291850	3	1.5	1060	9711	1	0	0	3	7	1060
229500	3	1	1780	7470	1	0	0	3	7	1050
323000	3	2.5	1890	6560	2	0	0	3	7	1890
662500	3	2.5	3560	9796	1	0	0	3	8	1860
468000	2	1	1160	6000	1	0	0	4	7	860
310000	3	1	1430	19901	1.5	0	0	4	7	1430
400000	3	1.75	1370	9680	1	0	0	4	7	1370
530000	5	2	1810	4850	1.5	0	0	3	7	1810
650000	4	3	2950	5000	2	0	3	3	9	1980
395000	3	2	1890	14040	2	0	0	3	7	1890

Tahapan Pemodelan



Latih Machine Learning Model



Data training:
Oversampling/Undersampling,
Standardize*,
Remove Outlier*
Feature Transformation

Data testing:
Standardize based on fitted in data
training*
Feature Transformation

**) Required for linear regression*

***) We do this after train/test split
to avoid data leak*

*Buat model ML
sesuai task*

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning

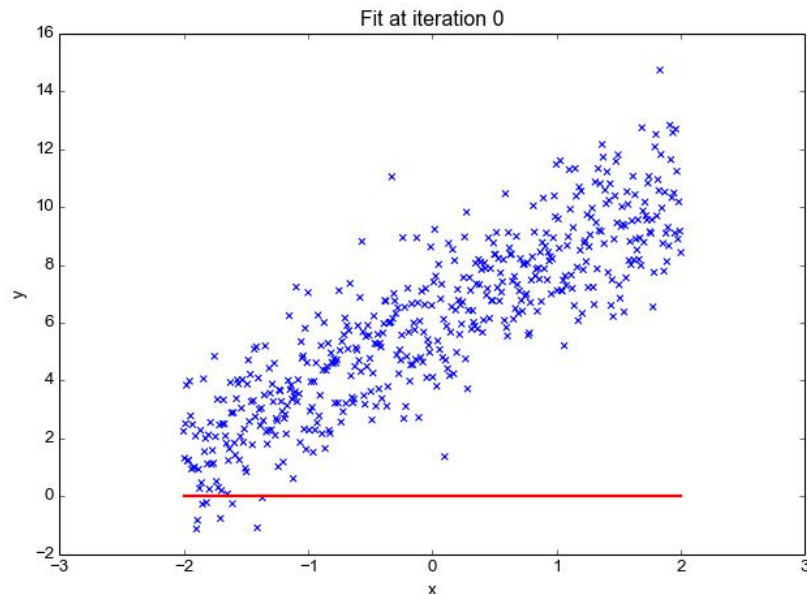


Hands On - Other Regression Algorithm

Main Idea & Intuition

Prediksi: Melakukan estimasi nilai Y berdasar nilai X

Source: [Statquest](#)



Linear regression adalah salah satu algoritma sederhana untuk melakukan regresi.

Menggunakan least-squares untuk **membuat garis linear yang fit pada data** (Hanya untuk tipe data numerik)

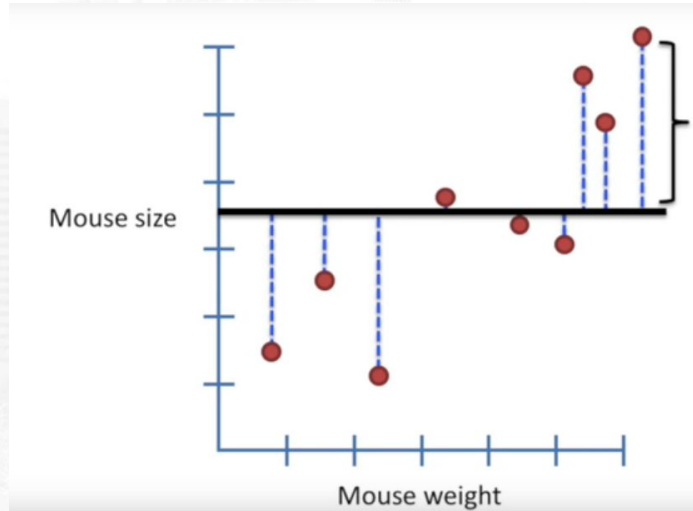
Bagaimana caranya?

Menemukan nilai minimum dari:

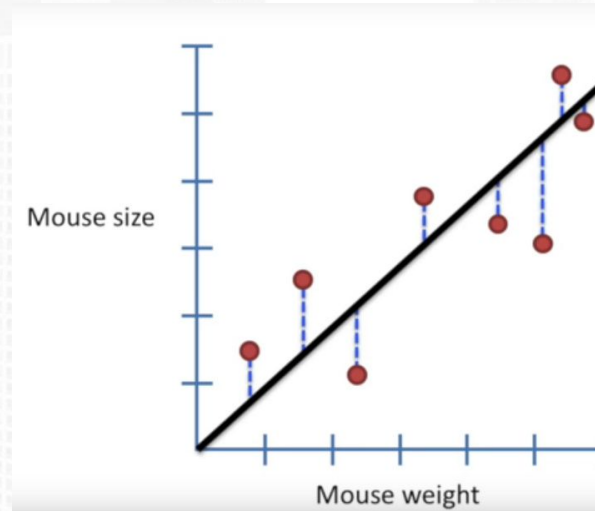
$$\sum_{i=1}^M (y_i - \hat{y}_i)^2$$

untuk setiap kombinasi garis linear

Menghitung jarak dari setiap kombinasi garis linear



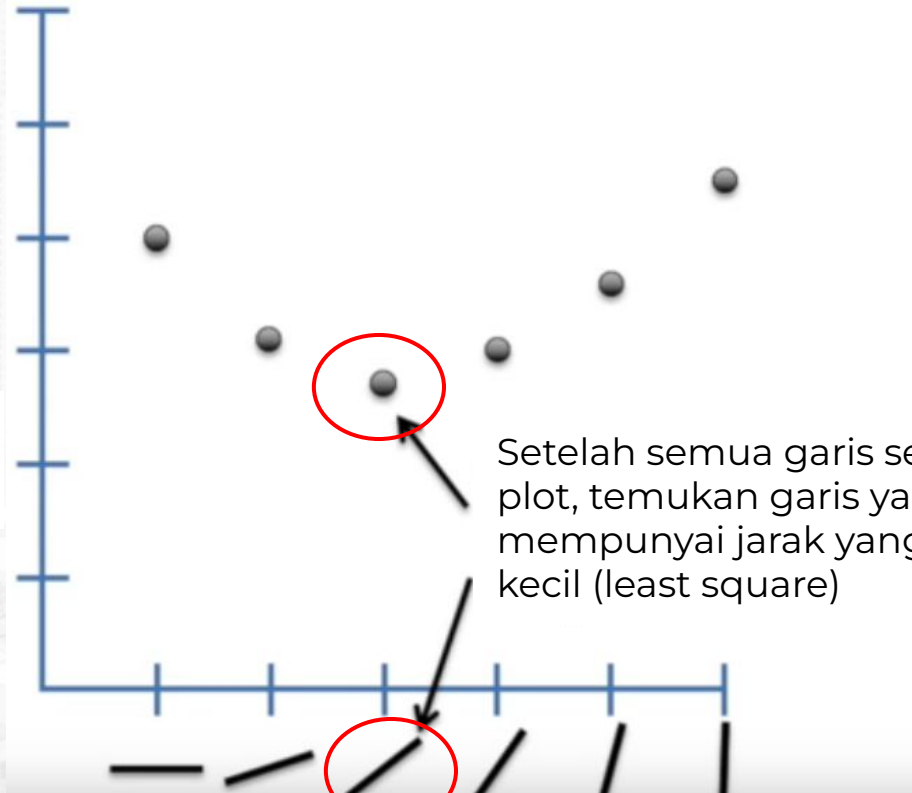
Berapa total
jaraknya?



Berapa total
jaraknya?

Total jarak
data point
dan garis

$$\sum_{i=1}^M (y_i - \hat{y}_i)^2$$



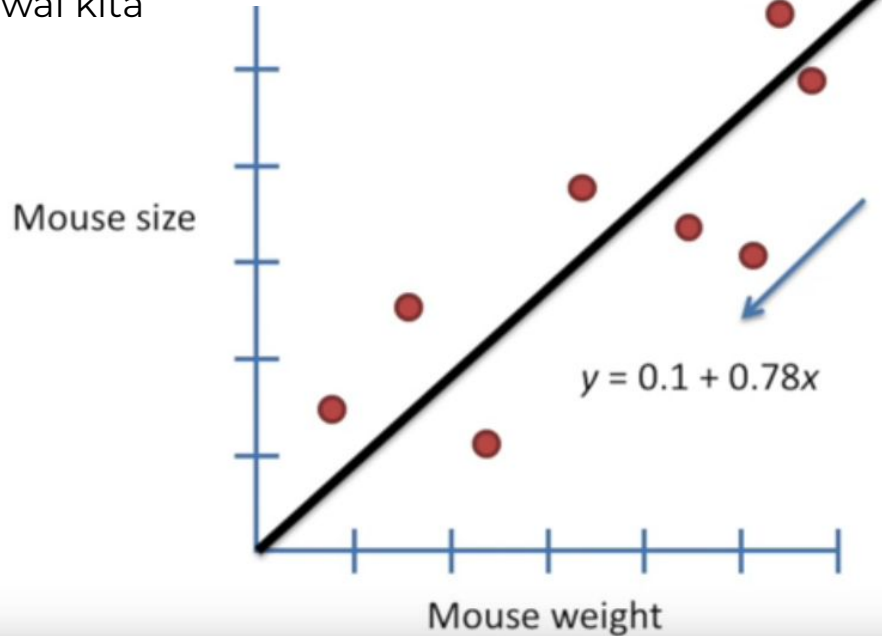
Setelah semua garis selesai di
plot, temukan garis yang
mempunyai jarak yang paling
kecil (least square)

Kombinasi garis linear

Jargon Alert!

jarak antara garis dan
data point yang paling
kecil disebut juga
dengan **least square**

Setelah menemukan garis dengan jarak yang paling kecil (least squares), gunakan garis tersebut sebagai hasil dari linear regression pada data awal kita



Persamaan garis linear yang terbentuk. ada 2 parameter yang dihitung di linear regression.

0.1 (intercept) dan 0.78 (coefficient)

Coefficient di-interpretasikan seberapa besar perubahan nilai Y jika nilai X bertambah

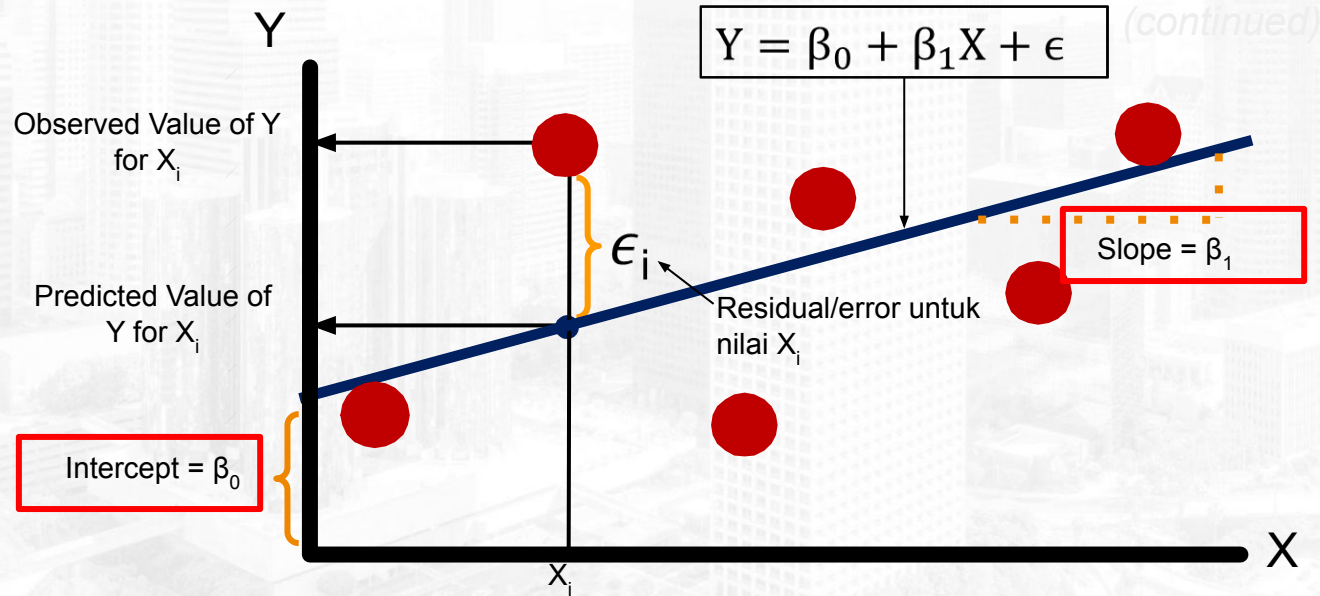
Linear Regression

- Variabel dependent (Y) nilainya akan berubah secara konstan ketika nilai dari variabel independen (X) naik atau turun
- Asumsikan bahwa hubungan antara variabel X dan Y linear

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Y : Dependent variable
 β_0 : Intercept
 β_1 : Slope / Coefficient
 X = Independent variable
 ϵ : Random error (Noise)

Simple Linear Regression



Evaluasi: MAE, RMSE, R²

- **Mean Absolute Error (MAE)** $MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$

- **Root mean squared error (RMSE)** $RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$

- **R²**: berkisar antara 0-1, mengindikasikan seberapa besar variabel independen mempengaruhi variabel dependen. semakin mendekati angka 1, model semakin baik.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

\hat{y} - predicted value of y
 \bar{y} - mean value of y

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multiple Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



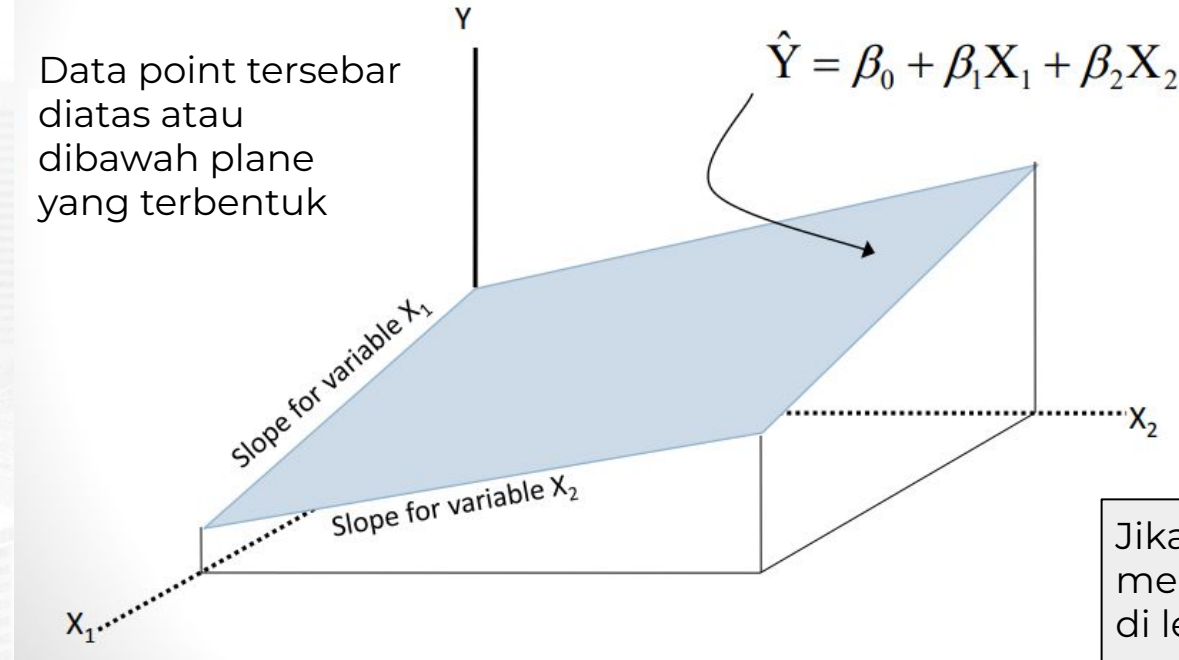
Hands On - Other Regression Algorithm

Untuk kasus di dunia nyata, dapatkah kita hanya menggunakan 1 variabel saja?

Kita dapat menggunakan lebih dari 1 variabel:
Multivariate Linear Regression

Multivariate Linear Regression

Data point tersebar
diatas atau
dibawah plane
yang terbentuk



Jika di 2D, kita
mengenal istilah garis,
di lebih dari 2D, kita
menggunakan istilah
plane

Multivariate Linear Regression

- Menggunakan lebih dari satu feature (X), dan melakukan prediksi untuk target (Y)
- Pada kenyataannya, di dunia nyata kita menggunakan banyak variabel

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i$$

Y : Dependent variable

β_0 : Intercept

β_i : Slope for X_i / coefficient

X = Independent variable

```

3 features = ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'zipcode']
4 complex_model_1 = linear_model.LinearRegression()
5 complex_model_1.fit(train_data_dm[features], train_data_dm['price'])
6
7 print('Intercept: {}'.format(complex_model_1.intercept_))
8 print('Coefficients: {}'.format(complex_model_1.coef_))

```


Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Coefficient As Feature Importance

Contoh Hasil Regresi:

Target:

Revenue (y)

Feature:

#visit (x_1)

#clicks (x_2)

Gender (x_3)

Contoh persamaan linear:

$$y = 0.1 + 0.7x_1 + 0.4x_2 + 0.0001x_3$$

Coefficient dapat diartikan sebagai feature importance dan dapat digunakan untuk:

1. Derive Business Insight

Contoh insight: Feature x_1 (#visit) berperan besar untuk meningkatkan revenue

Contoh rekomendasi: Perbanyak #visit dari user untuk meningkatkan awareness dan rekomendasi agar transaksi dan generate revenue, contohnya dengan gamifikasi

2. Feature Selection

Feature x_3 memiliki coefficient sangat kecil; 0.0001;

- Pertimbangkan untuk hapus jika featuranya jika menggunakan Linear Regression
- Jika featuranya kategorikal, gunakan algoritma lain untuk handle non-linear data

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Penting untuk diingat

Kapan lebih baik menggunakan Linear Regression dan Regularizationnya?

Linear regression harus memenuhi beberapa asumsi untuk membuat model tetap robust:

1. Hubungan Feature & Target linear
2. Hindari outlier
3. Setiap feature mempunyai distribusi normal, dengan skala yang sama
 - a. Setiap algoritma yang basisnya perhitungan jarak, penting untuk dilakukan scaling



Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Feature bedrooms & sqft_living tidak memiliki skala yang sama

Target

Feature

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above
221900	3	1	1180	5650	1	0	0	3	7	1180
538000	3	2.25	2570	7242	2	0	0	3	7	2170
180000	2	1	770	10000	1	0	0	3	6	770
604000	4	3	1960	5000	1	0	0	5	7	1050
510000	3	2	1680	8080	1	0	0	3	8	1680
257500	3	2.25	1715	6819	2	0	0	3	7	1715
291850	3	1.5	1060	9711	1	0	0	3	7	1060
229500	3	1	1780	7470	1	0	0	3	7	1050
323000	3	2.5	1890	6560	2	0	0	3	7	1890
662500	3	2.5	3560	9796	1	0	0	3	8	1860
468000	2	1	1160	6000	1	0	0	4	7	860
310000	3	1	1430	19901	1.5	0	0	4	7	1430
400000	3	1.75	1370	9680	1	0	0	4	7	1370
530000	5	2	1810	4850	1.5	0	0	3	7	1810
650000	4	3	2950	5000	2	0	3	3	9	1980
395000	3	2	1890	14040	2	0	0	3	7	1890

Penting untuk diingat

Kapan lebih baik menggunakan Linear Regression dan Regularizationnya?

Linear regression harus memenuhi beberapa asumsi untuk membuat model tetap robust:

1. Hubungan Feature & Target linear
2. Hindari outlier
3. Setiap feature mempunyai distribusi normal, dengan skala yang sama
 - a. Setiap algoritma yang basisnya perhitungan jarak, penting untuk dilakukan scaling
4. Hindari multikolinearitas (korelasi antar feature yang tinggi)
 - a. Hapus salah satu feature yang punya korelasi tinggi



Multikolinearitas pada Linear Regression

Jika terdapat hasil correlation analysis dibawah ini:

#visit	1		
#click	0.9 <small>(multikolinearitas)</small>	1	
revenue	0.7	0.8	1
	#visit	#click	revenue

- Feature #visit dan #click mempunyai correlation score yang tinggi satu sama lain. Jika #visit memiliki error yang tinggi di model linear regression, dapat dipastikan #click juga akan memiliki error yang tinggi juga karena korelasi 0.9.
- Error pada linear regression menjadi double ⚡

Penting untuk diingat

Kapan lebih baik menggunakan Linear Regression dan Regularizationnya?

Linear regression harus memenuhi beberapa asumsi untuk membuat model tetap robust:

1. Hubungan Feature & Target linear
2. Hindari outlier
3. Setiap feature mempunyai distribusi normal, dengan skala yang sama
 - a. Setiap algoritma yang basisnya perhitungan jarak, penting untuk dilakukan scaling
4. Hindari multikolinearitas (korelasi antar feature yang tinggi)
 - a. Hapus salah satu feature yang punya korelasi tinggi



Kalau ada kendala seperti diatas bagaimana?
Gunakan algoritma lainnya

algoritma lainnya. Beberapa detail akan dibahas di modul selanjutnya

Decision Tree

```
from sklearn.tree import DecisionTreeRegressor  
dt = DecisionTreeRegressor()  
dt.fit(x, y)
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor  
rf = RandomForestRegressor()  
rf.fit(x, y)
```

Support Vector Regressor

```
from sklearn.svm import SVR  
svr = SVR()  
svr.fit(x, y)
```


Topik Supervised Learning: Regression

- | | | | |
|-------------------------------------|-------------------------------------|--------------------------|---------------------------------------|
| <input checked="" type="checkbox"/> | Review Supervised Learning | <input type="checkbox"/> | Hands On Linear Regression - Part 2 |
| <input checked="" type="checkbox"/> | Linear Regression | <input type="checkbox"/> | Regularization |
| <input checked="" type="checkbox"/> | Multivariate Regression | <input type="checkbox"/> | Hands On - Regularization |
| <input checked="" type="checkbox"/> | Feature Importance | <input type="checkbox"/> | Hyperparameter Tuning |
| <input checked="" type="checkbox"/> | Asumsi Linear Regression | <input type="checkbox"/> | Hands On - Hyperparameter Tuning |
| <input type="checkbox"/> | Hands On Linear Regression - Part 1 | <input type="checkbox"/> | Hands On - Other Regression Algorithm |

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Implementasi di python



Implementasi di python : Model Validation

```
from sklearn.model_selection import train_test_split  
Xtrain, Xtest, ytrain, ytest = train_test_split(x,y,test_size=1/3, random_state=42)
```

Split train & test set

Implementasi di python : Fit Model

```
from sklearn.model_selection import train_test_split  
Xtrain, Xtest, ytrain, ytest = train_test_split(x,y,test_size=1/3, random_state=42)
```

Split train & test set

```
from sklearn.linear_model import LinearRegression# import linear regression dari sklearn  
regressor = LinearRegression() # inisiasi object untuk regression dengan nama regressor  
regressor.fit(xtrain, ytrain) # fit model regression dari data train
```

Latih model pada
train set

Implementasi di python : Prediksi Test Set

```
from sklearn.model_selection import train_test_split  
Xtrain, Xtest, ytrain, ytest = train_test_split(x,y,test_size=1/3, random_state=42)
```

Split train & test set

```
from sklearn.linear_model import LinearRegression # import linear regression dari sklearn  
regressor = LinearRegression() # inisiasi object untuk regression dengan nama regressor  
regressor.fit(Xtrain, ytrain) # fit model regression dari data train
```

Latih model pada train set

```
y_pred = regressor.predict(Xtest) # prediksi data test
```

Prediksi label test set

Implementasi di python : Evaluasi Hasil Prediksi

```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, ytrain, ytest = train_test_split(x,y,test_size=1/3, random_state=42)
```

Split train & test set

```
from sklearn.linear_model import LinearRegression # import linear regression dari sklearn
regressor = LinearRegression() # inisiasi object untuk regression dengan nama regressor
regressor.fit(Xtrain, ytrain) # fit model regression dari data train
```

Latih model pada train set

```
y_pred = regressor.predict(Xtest) # prediksi data test
```

Prediksi label test set

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
print(mean_absolute_error(ytest, y_pred)) # MAE
print(mean_squared_error(ytest, y_pred, squared=False)) # RMSE
print(r2_score(ytest, y_pred)) # R2 score
```

Evaluasi Test Set

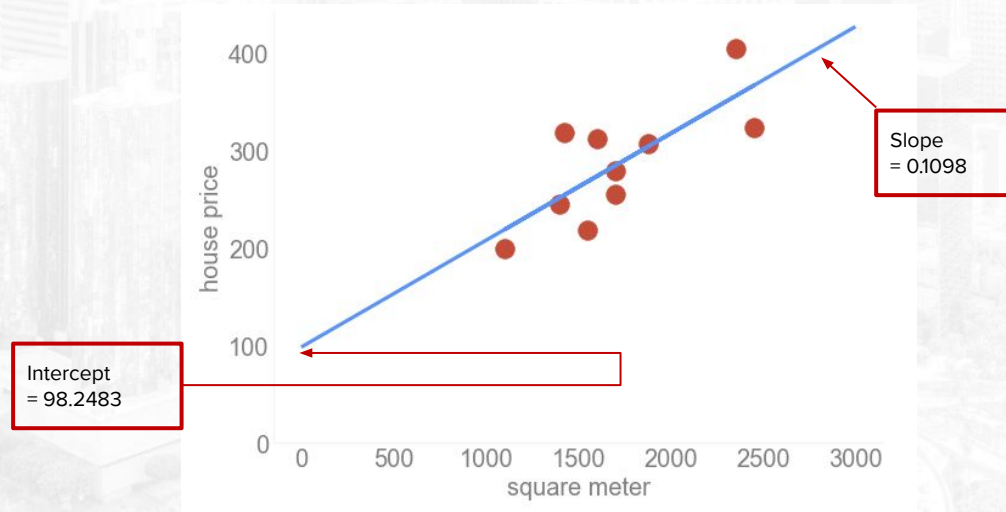
Output Hasil Evaluasi:

```
MAE: 0.02
RMSE: 0.03
R2 score: 0.61
```

Implementasi di python : Coefficient & Intercept

```
print('Coefficients: \n', regressor.coef )# The slope
print('Intercept: \n', regressor.intercept_)# The Intercept
```

Melihat nilai slope & intercept



Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning

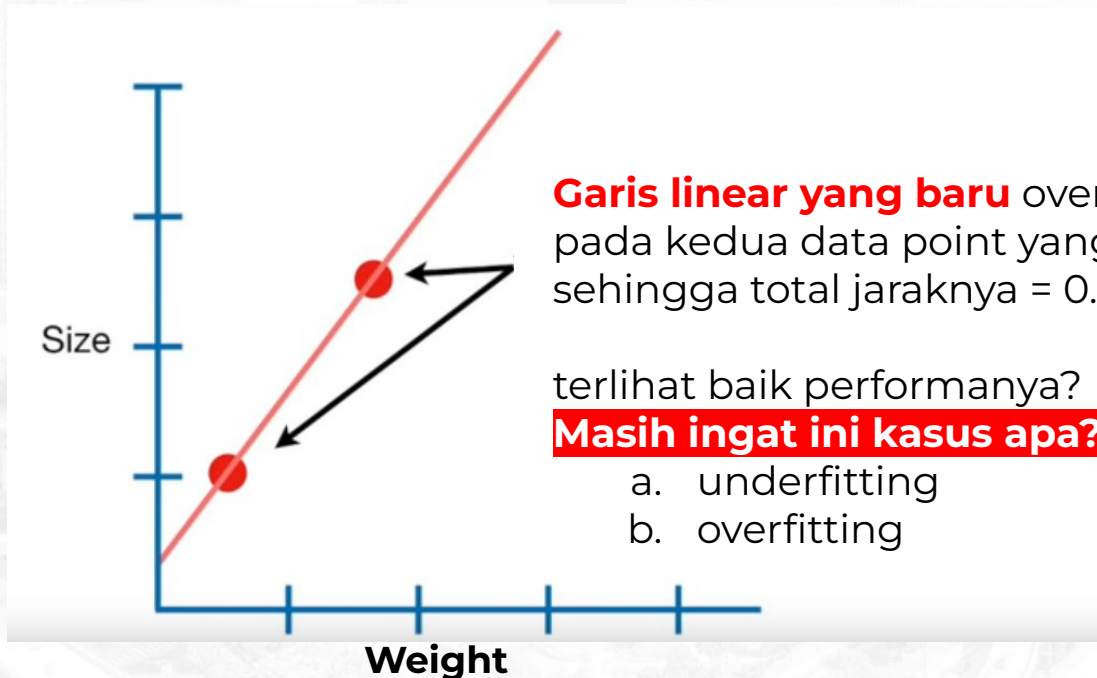


Hands On - Other Regression Algorithm

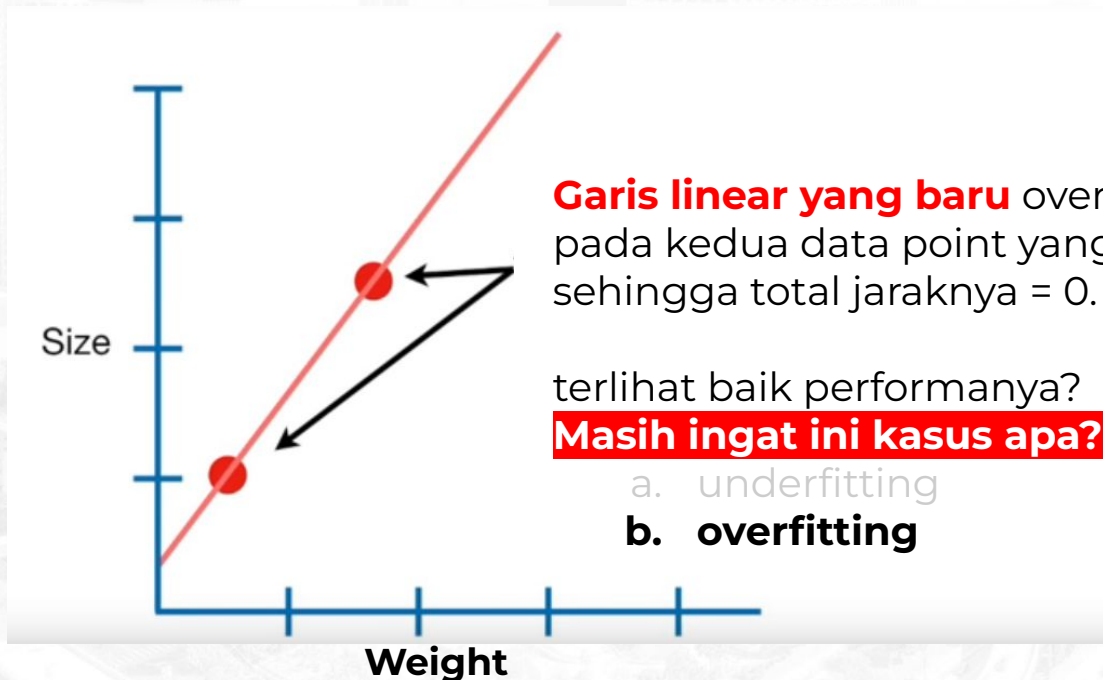
Sebuah Kasus:

Bagaimana jika kita hanya mempunyai sedikit data untuk training?

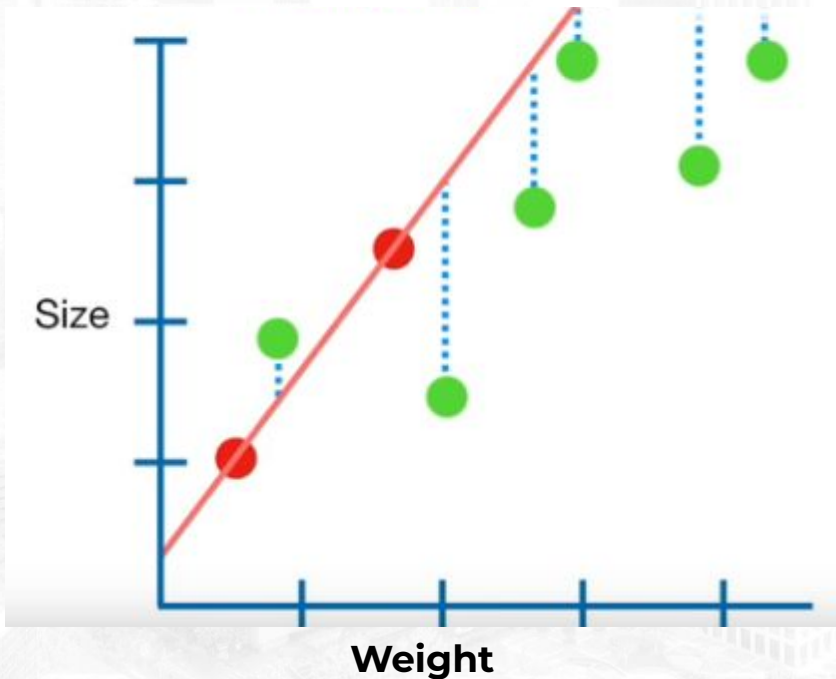
Nah, bagaimana jika kita hanya mempunyai 2 data point saja untuk membuat garis linear?



Nah, bagaimana jika kita hanya mempunyai 2 data point saja untuk membuat garis linear?



Jika **data point merah** adalah **data training**,
dan
data point hijau adalah **data testing**,

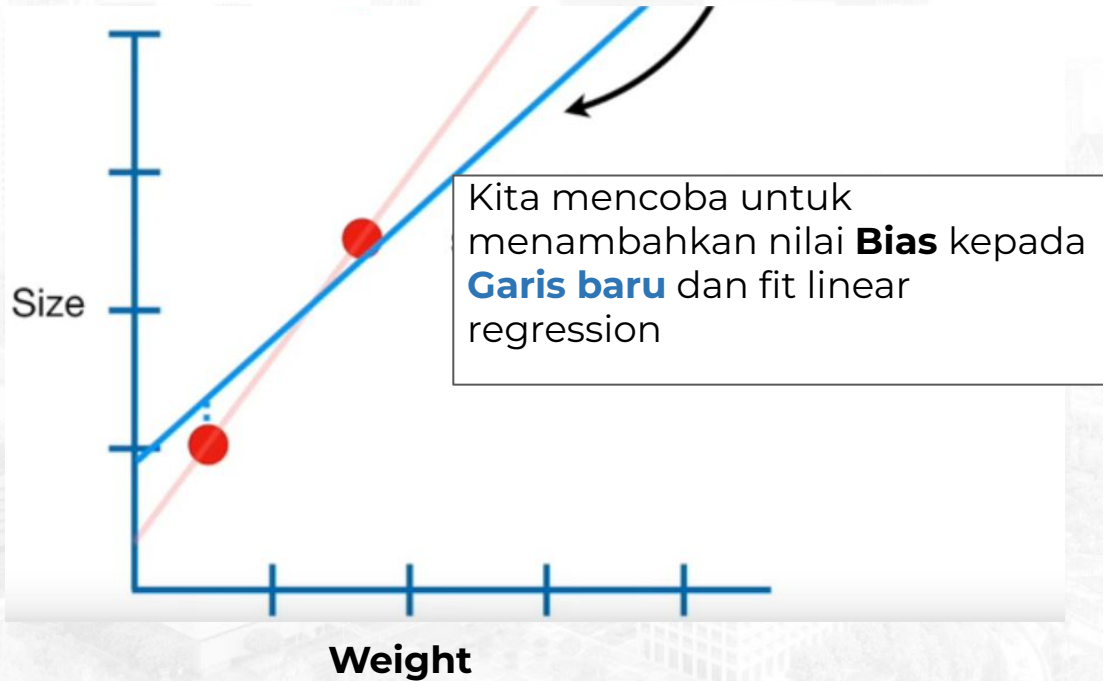


Total jarak antara **2 data point** pada **data training** sangat kecil, pada kasus ini adalah 0

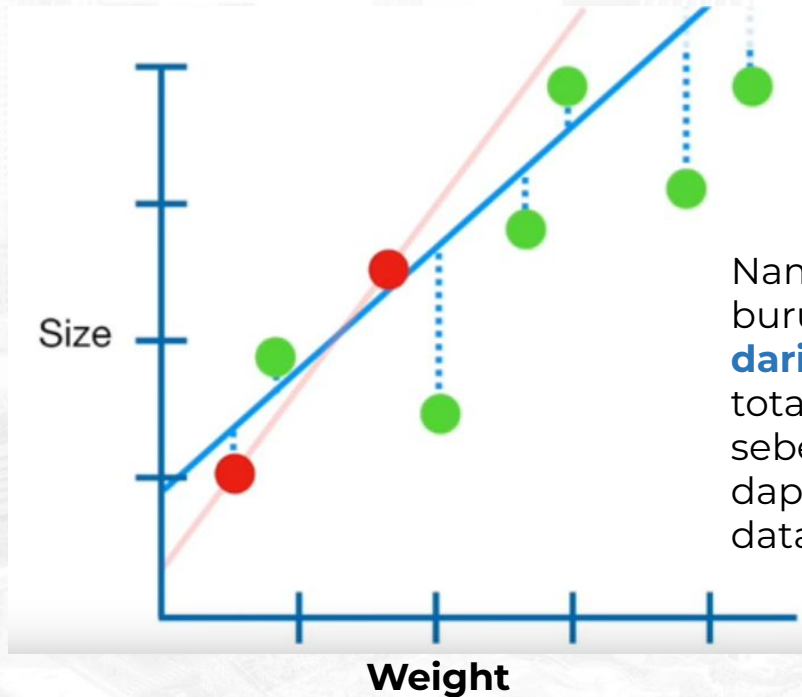
Namun, total jarak pada **data testing** terlihat cukup besar, benar?

Untuk mengatasi hal tersebut, dapat digunakan
Regularization

Ide utama dari **Regularization** adalah untuk membuat sebuah **Garis baru** yang tidak terlalu fit dengan **Data training**, namun diharapkan memiliki hasil yang lebih bagus dengan **Data testing**.



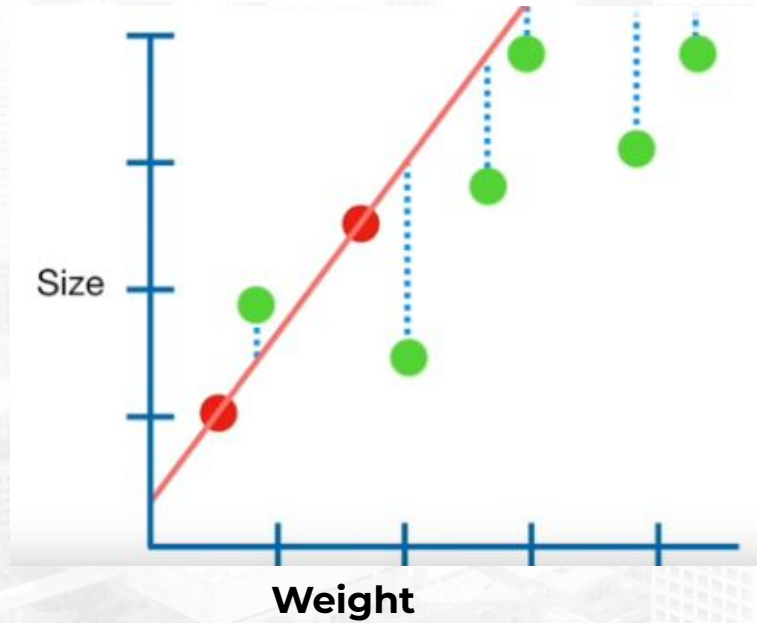
Dengan menambahkan **Bias**, otomatis kita memperbesar error pada **data training**, dan membuat hasilnya lebih buruk dari sebelumnya pada **data training**.



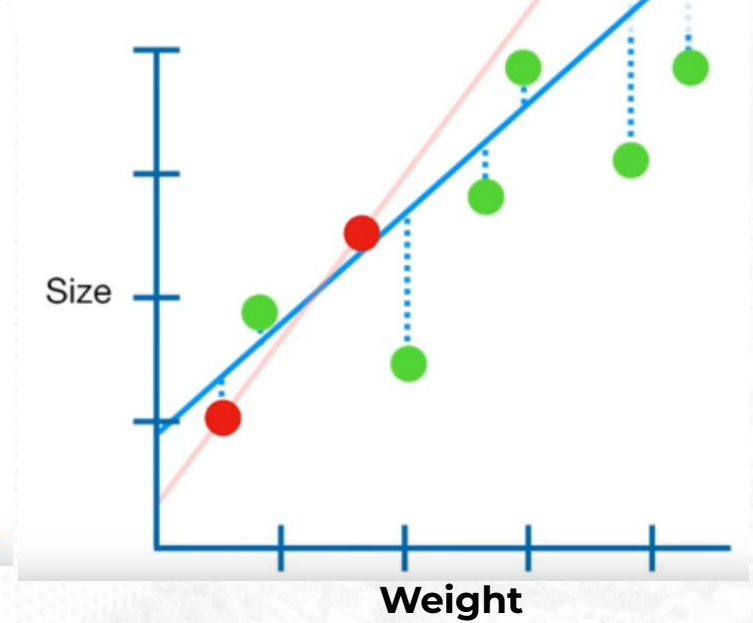
Namun, walaupun hasilnya lebih buruk pada **data training**, hasil **garis dari regularization** menghasilkan total jarak yang lebih kecil dari sebelumnya, yang mana modelnya dapat digunakan lebih general untuk dataset lainnya

Mari kita bandingkan lagi,

**Sebelum menggunakan
regularization**

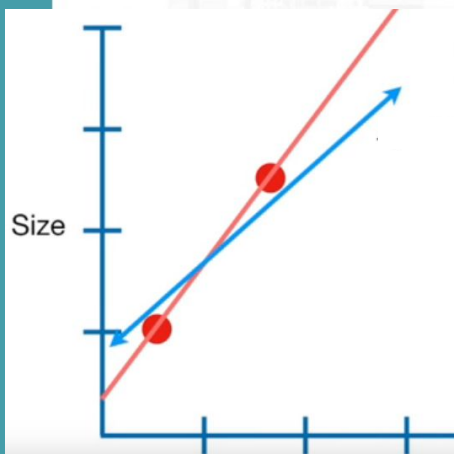


**Sesudah menggunakan
regularization**

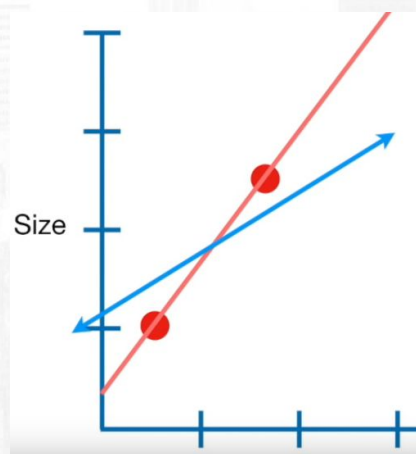


Pada regularization, terdapat parameter bernama λ (atau alpha pada sklearn). Parameter ini mempengaruhi seberapa besar nilai bias yang diaplikasikan.

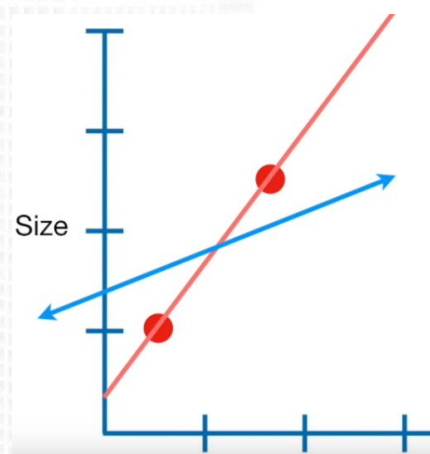
Nilai coefficient terus mengecil namun tidak pernah mencapai nilai 0



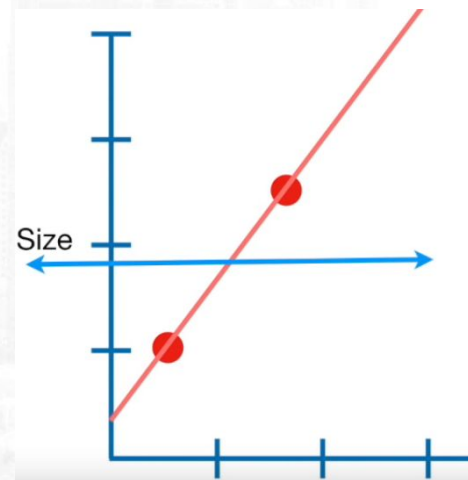
$\lambda = 1$



$\lambda = 2$



$\lambda = 3$



$\lambda = 10000$

Untuk memilih nilai λ yang tepat, kita melakukan **tuning hyperparameter** λ untuk menentukan hasil mana yang memiliki nilai variance yang paling kecil

kita akan membahas tentang tuning hyperparameter pada akhir topik

Regularization Types

- **Ridge Regression (L2 Norm)**

Performanya lebih baik jika banyak **variabel yang berkaitan** terhadap targetnya

- **Lasso Regression (L1 Norm)**

Performanya akan lebih baik dalam menurunkan variance yang mengandung banyak **variabel yang tidak terlalu berkaitan** terhadap targetnya

- **Elastic Net**

Kombinasi antara Ridge dan Lasso, mengatasi kekurangan satu sama lainnya. Namun, pemilihan parameter λ menjadi semakin kompleks

Jika kita mengingat kembali persamaan garis linear sebelumnya:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Ridge Regression menambahkan penalty L2-norm

$$Y = \beta_0 + \beta_1 X + \epsilon + \text{L2 Penalty}$$

$$Y = \beta_0 + \beta_1 X + \epsilon + \lambda * \beta_1^2$$

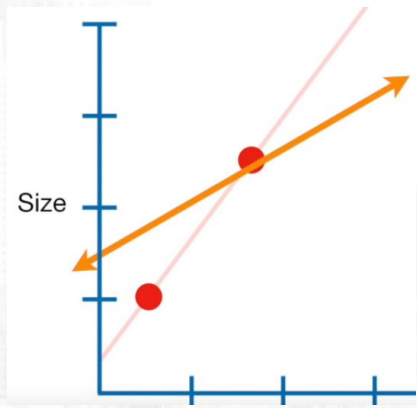
Lasso Regression menambahkan penalty L1-norm

$$Y = \beta_0 + \beta_1 X + \epsilon + \text{L1 Penalty}$$

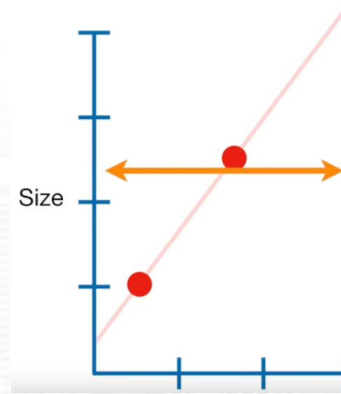
$$Y = \beta_0 + \beta_1 X + \epsilon + \lambda * |\beta_1|$$

menentukan
seberapa besar nilai
penalty-nya.

Perbandingan hyperparameter λ (atau alpha pada sklearn) pada **Lasso Regression**



$\lambda = 10$



$\lambda = 10000$

Nilai slope turun terus menerus
semakin besar nilai λ
nilainya bisa mencapai nilai 0
(tegak lurus terhadap sumbu y)

Artinya apa jika nilai slope 0?

Variable tersebut tidak akan ada pengaruhnya terhadap prediksi dengan linear regression. Ingat fungsi garis linear regression yang mengalikan variable dengan nilai slope

$$Y = \beta_0 + \boxed{\beta_1 X} + \epsilon + \lambda * |\beta_1|$$

Contoh: Bayangkan untuk melakukan prediksi size seekor tikus, kita memiliki variabel sebagai berikut:

- Weight
- Jenis kelamin
- Ukuran kantung
- Usia Pemelihara

Kedua variabel ini masih masuk akal

Bagaimana dengan kedua variabel ini?

Ketika kita menggunakan **Ridge regression**,

- Ukuran kandang
- Usia Pemelihara



Ketika kita meningkatkan nilai λ , nilai dari kedua variable ini akan mengecil hingga mendekati 0. tapi masih ada efeknya karena belum mencapai 0.

- Weight
- Jenis kelamin



Kedua variable ini akan sedikit menyusut nilainya seiring naiknya nilai λ

Namun, Jika kita menggunakan **Lasso regression**,

- Ukuran kandang
- Usia Pemelihara

→ Ketika kita meningkatkan nilai λ , nilai dari kedua variable ini akan mengecil hingga 0 sampai tidak ada efeknya lagi di perhitungan.

- Weight
- Jenis kelamin

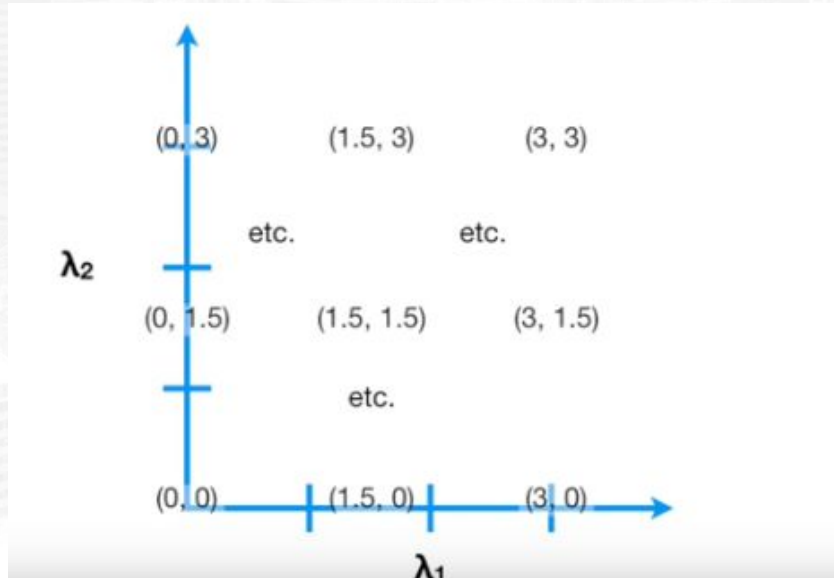
Kita akan mendapatkan hasil perhitungan yang lebih baik

→ Kedua variable ini akan sedikit menyusut nilainya seiring naiknya nilai λ

Elastic Net Regression - Why not use both?

Terdapat 2 lambda, yaitu untuk **lasso** dan **ridge**

$$Y = \beta_0 + \beta_1 X + \epsilon \quad + \text{L2 Penalty} + \text{L1 Penalty}$$



Perlu ditemukan kombinasi dari nilai λ_1 dan λ_2 untuk hasil yang optimal.

Ridge in Python

```
from sklearn.linear_model import Ridge  
model = Ridge()  
model.fit(X, y)
```

Lasso in Python

```
from sklearn.linear_model import Lasso  
model = Lasso()  
model.fit(x, y)
```

ElasticNet in Python

```
from sklearn.linear_model import ElasticNet  
model = ElasticNet()  
model.fit(x, y)
```

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Apa itu Hyperparameter?

- Variabel pada algoritma yang mempengaruhi hasil dari model

Contoh:

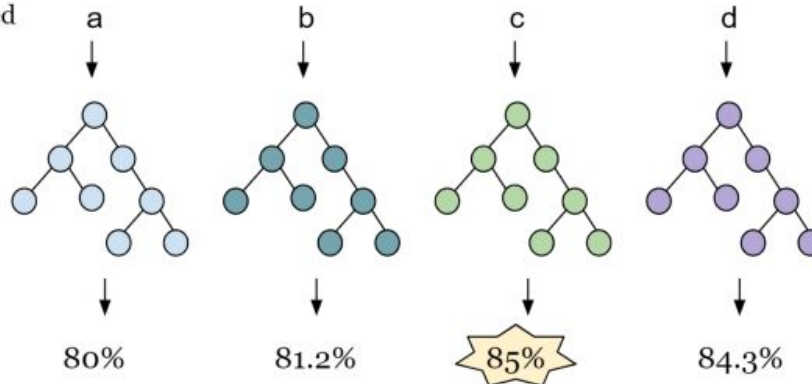
- Berapa nilai λ yang perlu digunakan di ridge regression?

Test_Hyperparameters = [a, b, c, d]

Hyperparameter used
to make the model

model

Accuracy on test set



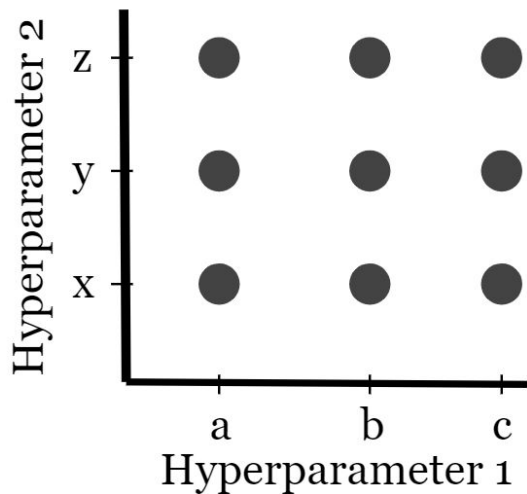
Solusi Umum: Grid Search

Grid Search

Pseudocode

```
Hyperparameter_One = [a, b, c]
```

```
Hyperparameter_Two = [x, y, z]
```

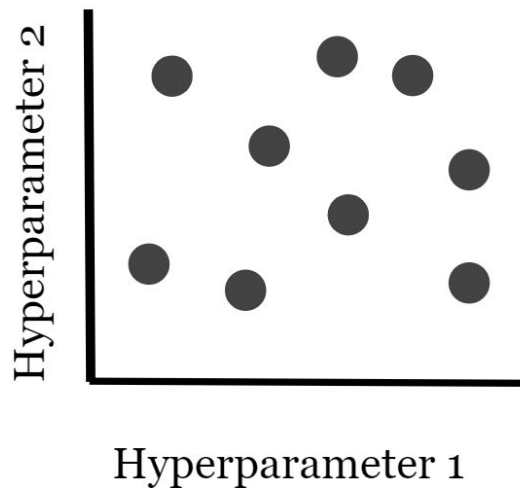


Random Search

Pseudocode

```
Hyperparameter_One = random.num(range)
```

```
Hyperparameter_Two = random.num(range)
```



Contoh

- Digunakan 2 hyperparameter untuk melatih algoritma ridge regression: alpha dan solver

alpha = [1,2,3,4]

solver = ['auto', 'lsqr', 'sag']



```
Ridge(alpha=1, solver='auto')
Ridge(alpha=2, solver='auto')
Ridge(alpha=3, solver='auto')
Ridge(alpha=4, solver='auto')
```

```
Ridge(alpha=1, solver='lsqr')
Ridge(alpha=2, solver='lsqr')
Ridge(alpha=3, solver='lsqr')
Ridge(alpha=4, solver='lsqr')
```

```
Ridge(alpha=1, solver='sag')
Ridge(alpha=2, solver='sag')
Ridge(alpha=3, solver='sag')
Ridge(alpha=4, solver='sag')
```

Akan dilatih 12 model ridge regression yang mempunyai performa error yang berbeda-beda. Model yang mempunyai error terkecil akan digunakan sebagai model akhir.

Implementasi di Python

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.linear_model import Ridge

# list dari hyperparameter
alpha = [0.01, 0.1, 1, 2] #  $\alpha$  /  $\lambda$ 
solver = ['lsqr'] # menggunakan least-square routine
hyperparameters = dict(alpha=alpha, solver=solver) # mengumpulkan semua hyperparameter pada dictionary

# Fit model
model = Ridge() # inisiasi model
clf = RandomizedSearchCV(model, hyperparameters, cv=5, scoring='r2') # randomize search dengan 5-fold cross-validation
best_model = clf.fit(xtrain, ytrain) # Latih model
```

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparamater Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm

algoritma lainnya. Beberapa detail akan dibahas di modul selanjutnya

Decision Tree

```
from sklearn.tree import DecisionTreeRegressor  
dt = DecisionTreeRegressor()  
dt.fit(x, y)
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor  
rf = RandomForestRegressor()  
rf.fit(x, y)
```

Support Vector Regressor

```
from sklearn.svm import SVR  
svr = SVR()  
svr.fit(x, y)
```

Topik Supervised Learning: Regression



Review Supervised Learning



Linear Regression



Multivariate Regression



Feature Importance



Asumsi Linear Regression



Hands On Linear Regression - Part 1



Hands On Linear Regression - Part 2



Regularization



Hands On - Regularization



Hyperparameter Tuning



Hands On - Hyperparameter Tuning



Hands On - Other Regression Algorithm